Chapter 8
Testing in the Cloud: Strategies, Risks and Benefits

Olumide Akerele, Muthu Ramachandran, and Mark Dixon

Abstract Testing in the cloud, commonly referred to as cloud testing, has revolutionised the approach adopted in traditional software testing. In the literal terms, it refers to testing Web applications in the "cloud" – leveraging a service provider's ready-made testing resources. The customer boycotts the hassle and expense of procurement, setup and maintenance of test environment setup on premise. Previously, accustomed solely with non-functional testing such as performance and load testing, recent advancements have made it possible to write test scripts and modify and automate test suites – all in the cloud environment. This chapter provides an in-depth overview of contemporary cloud testing, the types and its best practices. The benefits and risks are fully discussed with recommended methods to abate these risks. A methodological approach to govern an organisation migrating to cloud testing is also presented. A unique model, which shows the complex and dynamic interrelationship among active factors and their effect on the major project success factors in a cloud testing environment, is designed and presented. These project success factors include productivity, quality and cost. This model will help management to make strategic decisions on the adoption of cloud testing and the impact of their policy adoption on the productivity, quality and cost of software development projects.

Keywords Cloud computing • Cloud testing • Testing-as-a-Service • Agile software development

O. Akerele (✉) • M. Ramachandran • M. Dixon
School of Computing and Creative Technologies, Faculty of Arts, Environment and Technology, Leeds Metropolitan University, Headingley Campus, Caedmon Hall, LS6 3QS Leeds, UK
e-mail: o.akerele@leedsmet.ac.uk; m.ramachandran@leedsmet.ac.uk; M.Dixon@leedsmet.ac.uk

Z. Mahmood and S. Saeed (eds.), *Software Engineering Frameworks for the Cloud Computing Paradigm*, Computer Communications and Networks,
DOI 10.1007/978-1-4471-5031-2_8,

8.1 Introduction

Cloud computing has been a coveted buzzword in the computing industry and has been trending since the year 2005. Cloud computing is the technology behind the provision of software applications, data storage, computation and data access by service providers to their customers (businesses or individuals) – leveraging the Internet as the deployment medium [1] making the traditional infrastructure acquisitions necessary for on-premise computing operations unwarranted. The cloud service provider is typically located remotely, and the customer is charged proportionally to the amount of service or infrastructure usage.

Software development has been suggested to be improved by the introduction of cloud computing [2]. Apparently, every individual development phase and activity in the software projects can now be executed in the cloud; this includes coding, testing, deployment and maintenance of software projects. An area of viable potential research is the impact of the evolved synergy achieved from the adoption of cloud testing on a software development process methodology, particularly agile-based processes such as the Test-Driven Development (TDD).

Testing in the cloud, mostly referred to as cloud testing in this chapter, is the practice of carrying out the "testing" phase of the software development process in the cloud, hence preventing the need for the vast capital expenditure on acquiring infrastructure, licences and setup on customer site. The most popular applicability of cloud testing has been in carrying out performance and load testing where there is the vital need to generate multi-user traffic from various locations – which would ideally need numerous high-configuration servers for the traffic simulation. For example, a project needing to carry out load testing by hitting the test server with 30,000 users simultaneously will require a sizeable amount of infrastructure – with direct implicit soaring costs of test environment setup.

The ease and agility of the testing environment setup, reduced cost and maximum efficiency of the leveraged testing environment in cloud testing makes it an alluring option for both big- and small-sized software development teams. This chapter provides an overview of cloud testing and its various modes of deployment – with special emphasis on cloud testing in the context of non-functional Web applications testing in the cloud. Section 8.2 presents an overview of cloud testing, classes of cloud testing and the benefits, risks and a proposed best practice model for testing software applications in the cloud. Section 8.3 discusses the various ways in which cloud testing boosts the "agility" of a software development team. Section 8.4 describes a unique cloud testing causal loop model which summarises the activities, benefits and risks of adopting cloud testing.

In this chapter, the terms "software development organisation", "customer", "user" and "development team" are all used interchangeably. They all apply to the purchaser and adopter of the cloud testing service. Likewise, the terms "service provider" and "cloud vendor" are used in the same context, and both refer to the provider of the cloud test platform and services. In most parts of this chapter, cloud testing infers to performance, stress and load tests carried out in the cloud. However, it is implicitly specified at instances when it is used in the context of performing functional tests in the cloud.

8.2 Cloud Testing

Cloud testing is the carrying out of traditional testing practices using test resources situated in the cloud – made possible by the technology of virtualisation. This encompasses carrying out tests on both enterprise and Web applications in the cloud environment. Cloud testing can be classified under the three modes of cloud computing depending on the test activity requirements.

8.2.1 Types of Cloud Testing

Broadly, cloud computing is categorised under Infrastructure-as-a-Service (IaaS), Platform-as-a-Service (PaaS) and Software-as-a-Service (SaaS) [2]. The author has further classified cloud testing under these three categories of cloud computing as Cloud Testing Infrastructure-as-a-Service (CTIaaS), Cloud Testing Platform-as-a-Service (CTPaaS) and Cloud Testing Software-as-a-Service (CTSaaS). Figure 8.1 shows the types of cloud testing which are proposed and examples of the service providers and are defined and described as follows:

8.2.1.1 Cloud Testing Infrastructure-as-a-Service (CTIaaS)

This category provides organisations secured access to storage, hardware, networking components (including load balancers) and servers over the Internet for testing and development purposes. The infrastructure is the property of the service provider and is usually housed, run and maintained by the same. Customers pay for the amount of infrastructure needed for testing purposes, and this is maintained solely for the customer use by the service provider. All major testing activities are done on the customer site. Organisations have a high level of control over their instances and this category is deemed the most secured albeit it is the most capital intensive. The ease and low cost of racking up and tearing down the server makes it a very attractive option for organisations who are concerned about their data security. Customers are priced on a pay-as-you-go basis with the prices varying linearly with the number of "instances" of the server (Windows or Unix) and the software environment

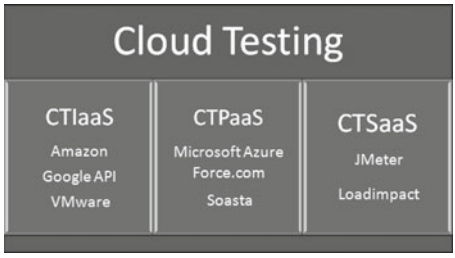

Fig. 8.1 Classifications of cloud testing

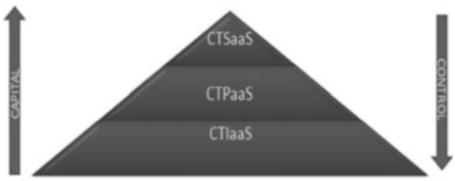

Fig. 8.2 Cloud testing stack

installed on them. The network traffic on the server as well as the volume of the data hosted on the server also influences the pricing. Typical examples of such service providers are VMware and Amazon. Figure 8.2 presents an overview of the three major modes of deploying cloud testing and their relative cost and control proportional relationships. The figure indicates CTIaaS is the most flexible category which provides organisations full control of their resources while CTSaaS is the least capital intensive as organisations do not require infrastructure or resource of their own for testing purposes.

8.2.1.2 Cloud Testing Platform-as-a-Service (CTPaaS)

CTPaaS provides a platform to development teams for functional testing purposes. This enables development teams leverage Cloud Integrated Development Environments (IDEs) with inbuilt unit frameworks to perform various functional tests and edit test scripts for test automation. CTPaaS vendors provide a subtle platform for operational facilities ranging from application development, testing and deployment environment. In other words, CTPaaS could be regarded as a platform for cloud-computing system development.

CTPaaS obliterates the need for the substantial capital that would otherwise be needed to set up a testing/development environment by helping to deliver the specific platform configurations through the Web browser interface. Hence, without any hardware or software investment, the platform of the service provider can be fully leveraged for software testing purpose. For instance, if development and testing requirements of a new application are Asp.net and SQL server database, an organisation would ideally need the following: VS.Net developer's licence, SQL server licences and deployment on a production server. The cost implication and configuration time of setup can all be done away with now! Also eliminated are the concerns for the staff and personnel expertise necessary to acquire and maintain the necessary infrastructure. CTPaaS enables users to select the testing requirement configurations via the Web browser interface presented by the service provider.

In some cases, such tests have to be written in the service provider's Domain-Specific Language (DSL). A typical example of this platform is the force.com with its proprietary programming language Apex Code. The major disadvantage of this is "Vendor Lock-In" [3] – which is literally the power the service provider possesses to lock customers into the service provision due to the customer's high cost of

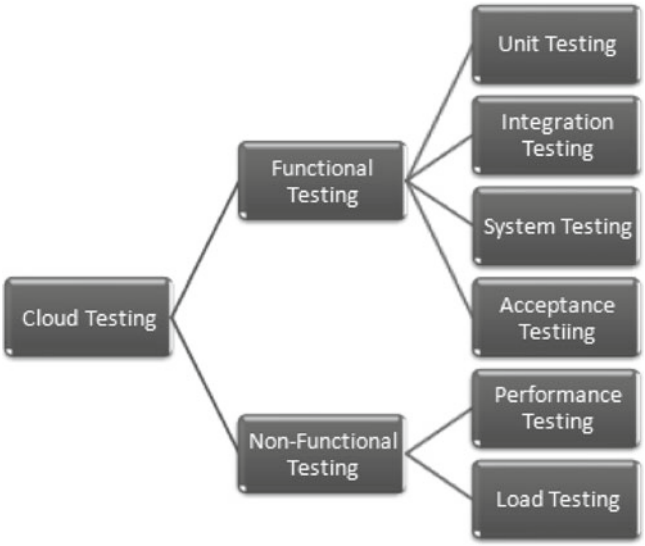

Fig. 8.3 Types of testing in the cloud

migrating to another provider or platform. The high costs of rewriting all the tests scripts in another programming language and the cost of training new personnel on a new platform make it an expensive option for customers to source other service providers.

8.2.1.3 Cloud Testing Software-as-a-Service (CTSaaS)

This is the most popular and adopted category of cloud testing and often illusively understood as the only application of cloud testing [4, 5]. Non-functional tests, particularly load and performance tests, are run on browsers offered by the service providers. Users enjoy the privilege of choosing the operating systems, browser types and versions, number of simultaneous Web traffic users as well as the various geographical locations of the mimicked generated traffic. This category allows Web application testing in the cloud by running the tests using real-life generated data traffic as input. The generated traffic is applied over the same communication channels as a Web browser, i.e. HTTP and HTTPS over ports 80 and 443, respectively. Examples of such platform are Loadimpact and JMeter. Figure 8.3 below broadly categorises the types of testing that are done in the cloud into functional and non-functional tests.

Cloud testing is used to apply solely to load and performance tests; recent advancements have seen it applied in numerous forms for functional testing such as unit tests, integration tests, system tests and user acceptance tests.

8.2.2 Economics of Cloud Testing

As in other business models, the Pareto principle (80-20 rule) can be effectively applied to software projects. The analogy hypothesises that software development teams running their own data centre infrastructure for testing utilise 80 % of their resources (capital, personnel, time, floor space) on acquiring and maintaining their own infrastructure, while 20 % is actually spent on the main value adding test activities of the project.

Cloud testing enables an organisation to deviate from this traditional "80-20" ratio described to a "20-80" – empowering organisations to swap 80 % of its resources on the actual core activities while 20 % of its resources are now spent on the non-value adding activities. With cloud testing, 80 % of the resources are spent on crucial activities like test script writing, unit testing, test automation and developing continuous improvement strategies for test process; 20 % of the resources are spent on less value adding activities like test environment setup, database configurations and browser installations.

Cloud testing has been suggested to provide up to 50–70 % testing cost savings – when appropriately adopted and integrated into software development organisations [4]. A significant amount of savings is made on infrastructure, licence purchase, storage mediums, multiple operating systems and experienced QA team. This also includes savings in labour for designing as well as building hardware and software platforms.

Fixed costs from the high investment on servers, network equipment and licence purchases are converted to variable costs as customers are charged on a "pay-as-you-go" basis. The evolved variable cost varies proportionally to the amount of the service provider's leveraged resource usage by the customer. Flat monthly/yearly fees charged to customers by service providers are no more the case, and customers hence do not pay for underutilised resources.

The reduction or eradication of the total cost of ownership is the most obvious attraction to cloud testing. The capital intensity of acquiring the infrastructure and platform for testing is avoided – making software development a more hospitable industry for start-ups and SMEs. The unused capital can then be diverted to fund more value adding and prioritised needs.

8.2.3 Benefits of Cloud Testing

The advantages of cloud testing to software development organisations and teams are systematically divided in three sections. The financial benefits to the organisation were discussed in Sect. 8.2.2. The third section is discussed later in this chapter in Sect. 8.3. The general impact of cloud testing is now discussed.

8.2.3.1 Improved Quality

Cloud testing reduces defects in web applications as compared to traditional on-premise testing [6]. There are two main factors responsible for this: the closeness of the simulation environment to the actual production environment making it easy to capture any bugs that will be encountered in real case scenario. Secondly, being the cloud vendor's area of expertise and core activity, they are more aware of the recent challenges faced, particularly security-wise, and are therefore able to put these into consideration during development and maintenance of their platforms.

8.2.3.2 Improved Accuracy

The cloud testing platform explored provides an environment with little or no variation from the actual production environment. Consequently, there is notable improvement in the predictability and accuracy of testing. There is also a higher degree of accuracy in the test. This makes the software quality more controllable, and this is a huge step for companies trying to adopt Six Sigma and achieve CMM levels 4 and 5 accreditation [3].

8.2.3.3 Waste Reduction

Another valuable benefit of cloud testing is that the organisation is charged on a pay-as-you-go basis by the cloud vendor. The scalability of the "rented" platform makes it possible for the organisation's price to vary linearly with the amount and time of cloud platform usage. This helps to reduce waste that occurs when infrastructure is redundant and helps the organisation to go "lean". This invention reduces the much valued time and effort spent by QA in creating the appropriate development and testing environment, hence greatly improving the Time-to-Market (TTM) of the developed software.

8.2.3.4 Improved Return-on-Investment (ROI)

Return-on-Investment will take a considerable leap when cloud testing is adopted appropriately. On-site QA team can now focus on improving quality and testing features instead of spending time acquiring, setting up and configuring infrastructure. They can now devote their time on process improvement activities. There is immense improvement in reliability of the tested application. The defect density and maintenance cost of the software would also subside consequently – improving the reliability of the software system. Substantial savings are made on maintenance costs on finding and fixing bugs on released software. The risk of losing customers

and competitive advantage due to complaints and damaged reputation is also reduced. Improvement in product quality improves customer satisfaction, retention and advocacy. Non-conformance cost is also reduced due to reduced resource to deal with customer complaints. As reported by a leading bank in 2009, their software projects adopted cloud testing and made overwhelming cost savings with a predicted ROI of 474 % over the following 3 years [7].

8.2.3.5 Green Testing

There is the ever-increasing need for industries to be environmentally responsible by going "green" and the IT industry is no exception. Cloud testing organisations enhance green testing. By sharing test resources in the cloud, businesses use IT resources solely on demand and this eliminate wastes by eradicating infrastructure idleness. In addition, organisations using cloud data centres can minimise energy use and deliver environmental savings in CO_2 by up to 80 % [8].

8.2.3.6 Easy Barrier to Exit

In the event the software development organisation decides to stop operation due to financial or strategic reasons, it is easier for the organisation to opt out as there is not much resources "locked-in", if at all any. This is unlike the difficulty to exit operations faced by an organisation that has made a lot on investment in acquiring these infrastructures, storage and operating systems to carry out its testing activities on site.

8.2.4 Best Practices for Migrating to Cloud Testing

Migrating to cloud testing has to be a systematic process otherwise it could turn out to be a fiasco. A shrewd and meticulous approach has to be adopted to avoid failure and reap the maximum benefits of cloud testing. The recommended sequential activities during the migration process are represented in Fig. 8.4.

Figure 8.4 depicts the diagrammatic flow of the best practices before migrating to cloud testing. They are explained below.

8.2.4.1 Cost-Benefit Analysis

This is the most crucial and fundamental process in the roadway to leveraging cloud test platform. It involves the preliminary feasibility study and an in-depth breakdown of the benefits and the cost associated with its adoption in the organisation. This is usually handled by a cloud broker who will be conducting full auditing of

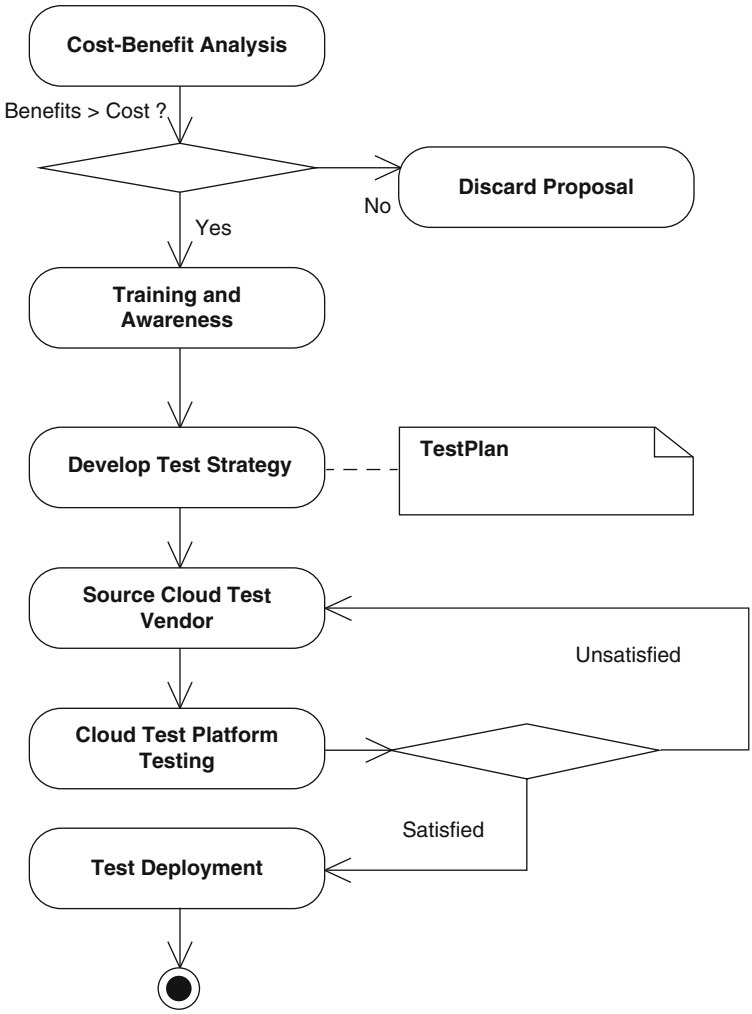

Fig. 8.4 Cloud testing migration model

the entire testing infrastructure requirements of the organisation. Ideally, a 2-year ROI analysis should be enough to give an insight into its viability. Some organisations however require a minimum of 5-year ROI projection. It is a common illusion that cloud-computing testing is generally cheaper. Ongoing intrinsic costs associated with cloud testing include the cost to support privacy regulation policy, cost to build auditing processes in the system and recovery service cost [8]. Before such conclusion could be reached, all the related cost associated should be considered holistically. It is easy for organisations to fall for "Management-by-Magazine" approach particularly with new computing innovations and cloud testing is one of them.

8.2.4.2 Training and Awareness

The transitional process should continue with the formal introduction of the relevant teams to cloud computing and testing. This can be classroom based or simply provide texts from the "dummies" series on cloud computing. This is to provide a foundational understanding on its applications and how they are applied to cloud testing. The major impetus for the need for cloud test migration in the organisation should be emphasised, and necessary feedback from the stakeholders at this stage is vital to the successful adoption of cloud testing.

8.2.4.3 Developing Cloud Testing Strategy

This should be developed beforehand and should be informative. It should be made available to the stakeholders to envisage the aims and objectives of the proposed initiative – while welcoming any feedback from the stakeholders. Vital constituents include goals of the initiative, infrastructure and resource requirements, types of tests (load testing, stress testing, security testing, functional testing) to be migrated to the cloud and anticipated risks with the corresponding mitigation techniques.

8.2.4.4 Sourcing Cloud Testing Vendors

After the test environment configuration requirements have been elicited, the next step is to find the most suitable cloud test provider that can satisfy the team's goals and objectives. Failure to secure an adept and reliable vendor will lead to inconsistent and erroneous testing, and this could turn out to be frustrating and regretful. The "ramp-up" and "tearing down" time to initiate cloud testing should also be considered when choosing the vendor. A good Service Level Agreement (SLA) must also be negotiated by the management before signing a contract. The barriers to easily change the provider are also paramount to make switching service providers easy and must be considered. Figure 8.5 summarises the selection process for the right cloud test vendor.

8.2.4.5 Cloud Test Platform Testing

Equally important is the testing of the cloud test environment to ensure the cloud vendor can perform to meet with the development team's goals. Adequate testing also helps to alleviate possible risks that may occur during testing. Vigorous testing performed should be aimed at determining the cloud environment's level of security, scalability, reliability and performance. These tests should be fully run before an agreement is signed with the cloud provider and before executing any tests. Other aims of testing the platform should focus on data governance, manageability, availability, latency, connectivity, regulatory compliance, uptime and privacy. There are

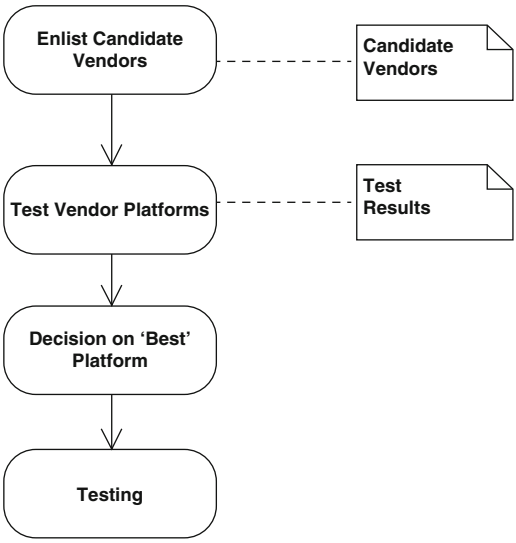

Fig. 8.5 Platform sourcing process

third-party tools such as Cloudstone and Cloud Harmony that can be utilised in testing the performance of the cloud test environment.

8.2.4.6 Test Execution

It is imperative that the test environment is configured correctly before test execution. Thirty percent of all defects are reported to be caused by wrong test environment configurations [5]. Following the success of the previous steps, test activities can now be securely carried out on the vendor's platform. A good practice is to start the initiation process with the more experienced testers who can quickly grasp the dissimilarities between cloud testing and the traditional on-premise testing – and use this to mentor the less experienced ones on the best practices in the new environment.

Following the successful migration of the testing activities to the cloud, activity monitoring, analysis and reporting must be continuously ensured. The performance of cloud testing must be evaluated constantly by the development organisation – particularly during real-time simulations. Productivity of cloud testing might be lower initially considering the learning process of getting used to the new test environment but will improve significantly with time if this approach is followed diligently. A major factor is the learning curve of the testers in learning the DSL of the service provider's platform (if any). Metrics such as defect density, test coverage and the likes should also be closely monitored – while being alert at any eyebrow-raising variability in the system performance. The SLA should be closely monitored to ensure the vendor is keeping to their part of the agreement. It is a good practice to hold regular meetings with the cloud test vendor to highlight any areas of concerns, risks or issues that might arise during the course.

Table 8.1 Risks introduced by cloud testing

Business risks	Project risks
Security	Non-conformance to organisational policy, strategy and methodology
Termination of service provision	Vendor's server breakdown/availability
Inaccurate cost-benefit analysis	Abandoning trusted legacy testing resources
Biased cloud brokerage	Internet connectivity
Industrial espionage	
Vendor lock-in	

8.2.5 Risk Assessment in Cloud Testing

Risks in leveraging cloud testing by the customer are bidirectional: while the customer transfers some risks during the process, they are also prone to certain risks from the vendor. Adopting cloud testing must therefore be meticulously planned with proactive measures to reduce the occurrence of the highlighted uncertainties [9].

For the purpose of clarity, these risks are classified into business risks and project risks. Business risks in this context are those that have a direct impact on the profitability and reputation of the organisation, while the project risks have a direct impact on the success of the software engineering project. Table 8.1 outlines the risks the customer is prone to when adopting cloud testing:

The risk items tabulated above are further explained in the following section.

8.2.5.1 Non-conformance to Organisational Policy, Strategy and Methodology

The testing processes of the service provider might not be following the principles governing the customer's organisation. Service providers usually have separate regulations governing their operations and infrastructure management. This might be difficult to verify even when specified in the SLA due to the limitation of the customer's involvement in cloud vendor's activities. This is unlike the on-premise sites where there is a governance system to ensure compliance. This can be mitigated by emphasising on compliance on the contract agreement and strictly outlining the organisations' specific policies.

8.2.5.2 Security

The major disgruntlement and concern in cloud testing revolves around security – especially when the user's sensitive data will be stored in the production environment sat in the cloud. The utilisation of a second party's platform creates an atmosphere

for paranoia because organisations usually cede control of the platform and data to the vendor. This is exacerbated when part of the cloud services is federated to a third party by the cloud vendor. A couple of infamous let-downs by cloud vendors that epitomise the above security concerns are:

- Amazon's "glitch" in April 2011 which was responsible for numerous Web sites' malfunction.
- Sony of Japan revealed that about 100 million of PlayStation customer accounts had been hacked.

This risk is abated by adoption of security testing tools and vigorous hacking techniques. A disaster recovery test also helps to have an insight into the reliability and dependability of the testing service provider. These tests should be ongoing and should also precede the test execution process as discussed earlier.

8.2.5.3 Industrial Espionage

This is also a major concern particularly when the testing activities are completely outsourced to a cloud test service provider. The cloud vendor's personnel could be easily "tapped" for information about the development organisation's product and be offered a reward for such unscrupulous act. For example, the cloud vendor's staff could trade in significant features of a product yet to be released to the rival organisations. The rival company could then strategically match or even better the feature and incorporate this into their similar product. This puts the development organisation in severe risk of losing their competitive advantage in such scenario. To reduce this risk, consequences of information divulgence must be reasonably severe, explicitly communicated and stated in a non-disclosure agreement offered to cloud vendor personnel.

8.2.5.4 Termination of Service Provision

In the event the service provider terminates service provision due to financial or strategic reasons, the user could potentially be in trouble if adequate provision and flexibility for adopting a "plan B" is not in place. Strategic decisions could be due to merger and diversification or simply the company's decision to concentrate on more profitable business activities. Service providers reserve the arbitrary right to terminate customer's account without notice due to policy violations; this however happens in exceptional cases. This risk can be reduced by checking the vendor's history to ensure it has been sustainable and fairly successful in operations for a reasonable length of time – to give some assurance on its reliability in the industry. Also, the signed contract should indicate a fair notice period for termination of contract by either party.

8.2.5.5 Inaccurate Cost-Benefit Analysis

Risk of unanticipated soar in operation cost associated with cloud testing is also a possibility when the cost-benefit analysis is not done correctly with all latent costs unravelled. This could make the decision of adopting cloud testing a regrettable one should the costs outweigh the benefits on the long run. These costs should include, if applicable, cost of writing or modifying the entire test cases should the cloud test platform require applications tested in its own discrete language. The opportunity cost of such activities should also be considered. The analysis and comprehensive cost projection should be cross-checked by a second cloud expert – who could be an insider so as to reduce cost.

8.2.5.6 Biased Cloud Brokerage

Cloud brokers act as intermediaries between cloud vendors and individuals or businesses purchasing there service. They help the potential users to understand their needs and source the possible best cloud service providers in return for an agreed premium. This involves the end-to-end audit of the entire incumbent testing process of the customer and the network infrastructure and usually putting up the cost-benefit analysis as well as the expected ROI necessary to make the business case for cloud testing. This consultancy service is usually costly but it is a vital process and must be done regardless if it is done in-house or outsourced. The challenge is actually in getting an unbiased broker because most cloud brokers usually have some sort of affiliation with specific cloud vendors and they are rewarded for their referrals. Hence, there is the tendency for brokers to always recommend adopting cloud testing as the best solution to customers even when that is not the case. Also, cloud brokers advocate the "best" cloud vendors to their customers, and these vendors are usually the ones they have affiliation with, not necessarily the best service providers to meet the customer's needs. Getting acceptable references from the broker's customers is a way of reducing this risk. Also, ensuring that the final decision of the choice of the vendor is in the user's hands helps to reduce getting "sold" to a cloud vendor.

8.2.5.7 Abandoning Trusted Legacy Testing Resources

Discarding the old testing infrastructure and resources poses a huge risk to an untested technology in an organisation. Though the on-premise testing requirements are quite expensive to maintain, replacing them with untested technology could be disruptive, more expensive and potentially risky. Retraining of staff on the new cloud vendor's system and also learning the service provider's DSL (if any) take a lot of time, and an initial reduced productivity should be expected. This risk can be controlled by avoiding the big bang approach to cloud test migration. Selecting a vendor that utilises the user's familiar application language is also very important and helpful.

8.2.5.8 Vendor Lock-In

Many cloud platform vendors, like force.com using Apex Code [10], offer services on their platform in their Domain-Specific Language (DSL) – making it difficult to move applications to another CTPaaS. Due to the high switching costs (time, effort and other resources as well as cost already spent on personnel learning the proprietary language) in migrating to a new platform, customers are tied down to the vendor. This gives the vendor the power to increase the subscription rates at any time, and this would have an adverse effect on the ROI. Google, another example of vendor that provides app engine CTPaaS in its proprietary language, recently announced a shocking increase of 100 % in their pricing which caused an intense backlash from users. The best way of eliminating this risk is avoiding lock-ins by all means. Choosing programming languages that are easier and faster to modify can also reduce the impact of this risk.

8.2.5.9 Cloud Vendor's Server Breakdown

The breakdown of the testing platform server entirely paralyses the testing activities of the customer. This affects not only the testing activities but all other activities dependent on the testing phase. Hence, the server needs to be up and running as well as being available at all times to prevent this period of no activity. Running availability tests before choosing the vendor can help reduce this risk.

8.2.5.10 Internet Connectivity

The success of the testing activity is fully reliant on the provision of fast, reliable, dependable and robust Internet connection. Necessary network infrastructure to provide this is a prerequisite to venturing into cloud testing with a service provider. This is because should the Internet connection fail, testing activities cannot be run on the vendor's platform and this could result in substantial loss as there will not be connection to the host server.

8.3 Impact of Cloud Testing on Software Development Agility

Agile software development values quick feedback to customers, collocation and easy collaboration between team members and customers. This reduces idleness by capitalising on the "just-in-time" approach of the development activities and flexibility to user requirements at any stage in the development process. Cloud testing ticks the fore-mentioned features to enhance the "agility" of a software development process and is therefore considered a good match – particularly to distributed agile teams. The relative impact of cloud testing on agile development projects is detailed below:

8.3.1 Reduced Time-to-Market (TTM)

Feedback is vital in agile processes. Cloud testing significantly reduces the test cycle times of software projects and consequently the deployment cycle. Cloud testing not only has a major impact on the TTM, it also improves the flexibility of the system to accommodate changes and requirement creep [5]. The type of agile methodology adopted also has an impact on the impact of cloud testing on the deployment cycle. For example, when TDD is adopted, concise and just enough code needed to design and test a function is written. With this, even more time is saved in coding and testing by avoiding extraneous coding and this will positively impact the TTM for the software.

8.3.2 Support for Geographically Dispersed Teams

Cloud testing eradicates the problem of proximity in dispersed development teams. It bridges the geographical distance between global teams – enabling easy interchange and handover of feature development among teams as if they were collocated. Teams can now collaborate globally with a self-defined user interface. The omnipresent accessibility of the cloud test platform via a common URL makes it possible for distributed teams to perform testing without geographical barriers and makes handing over easier.

8.3.3 Visibility and Accessibility

Teams can now collaborate in real time; this overcomes the problem of delay in hours or even a day in sending and receiving data between offshore and onshore colleagues. Every testing activity can be revealed and made visible in real time and accessed from anywhere via a custom URL for the organisation.

8.3.4 Support for Automated Testing

There is an increasing demand for automation in software development due to its significant time savings on the development cycle [5]. Automation is now being applied in development, testing and even software deployment. Cloud testing fully supports agility in testing by creating a welcoming platform for writing and importing automated scripts for functional testing. Cloud test platforms usually have plug-ins that allow the recording of the test activities to be analysed after test completion. Valuable time can now be spent investigating new possible bugs instead of

exhausting time on running repeated tests manually. Cloud testing also provides an environment to concurrently run tests with different configurations on the same machine. This produces financial savings due to customers being charged on time consumed for testing.

8.3.5 Requirement Volatility Support

Cloud testing encourages frequent changes to requirements as it provides testing accessibility anywhere and testing the modified system is made possible anywhere. The effect (estimated finish date) of the frequent changes in the requirements and requirement creep is offset by the reduced test cycle achieved by cloud testing. Also, the scalability of the system makes it possible to test each iteration feature as each iteration feature might require the ramping up and scaling down of the system requirements.

8.3.6 Bug Reproduction

Another vital importance of cloud testing is the quick ability to reproduce bugs for further analysis. This has been a major issue in traditional testing. There is a constant need to regenerate bugs that were detected in an environment for further investigation. During simulation, snapshots at the point of software failure could be taken to show the entire configurations when bugs are found – enabling testers to quickly revert to the configuration needed to reproduce the bugs and they can start debugging in no time.

8.3.7 Support for Test-Driven Development

Cloud testing creates synergy by being fully supportive of agile development techniques such as the TDD. Automatic unit tests are written to fail, pass and refactor – all in the cloud. This yields great time savings and fosters quick customer feedback while guaranteeing improvement in software quality.

8.3.8 Parallel Testing

Agility is improved by allowing tests to be run in various scenarios concurrently without having to test consecutively on the same physical machines. It also facilitates testing various components simultaneously. This increases throughput and

coverage while reducing test cost. It enables the tester to create different configurations such as the operating system database and storage to meet the actual production environment specification.

8.4 Cloud Testing Model

The cloud testing model described in this section diagrammatically summarises the entire cloud testing system – giving an instant high-level view of the processes, advantages, risks and consequences of actions within a cloud testing environment. All the components within the cloud test platform are modelled as interrelated factors in feedback loops. This model succinctly describes the variables, influencing factors and the dynamic influence of cloud testing activities in a software organisation. It shows the interrelationships and continuous nature of the actions performed in cloud testing and the resulting positive and negative impacts in the cycle.

The model provides a high-level general overview of cloud testing to management for decision-making purposes regarding adoption of cloud testing in the software organisation. The model can be used to trace the root causes of irregularities as well as improvements achieved when leveraging cloud testing platform. It provides an instant insight into the risks the potential users become susceptible to when testing applications in the cloud and also presents the opportunities. A significant portion of this model can also be applied to cloud computing in general. The positive and negative polarities indicate constructive and degenerative impact on the variables respectively. A number of assumptions have been made in constructing this causality model:

- Testing is fully automated.
- All other activities before development and after testing are performed on premise.
- Iteration-based development approach is adopted.
- The cloud vendor uses a DSL.

This dynamic model is presented in Fig. 8.6. The positive and negative polarities in the designed model indicate constructive and degenerative impacts on the directional variables respectively.

For brevity, a brief explanation of the major variables in this causal model diagram is explained below. The factors referenced in the model are italicised.

8.4.1 Productivity

The *Corporate IT Governance* in an organisation should encourage the provision of up-to-date training on cloud testing, thus improving the *organisation cloud awareness.* This earns the *stakeholder support* as well as better *understanding cloud*

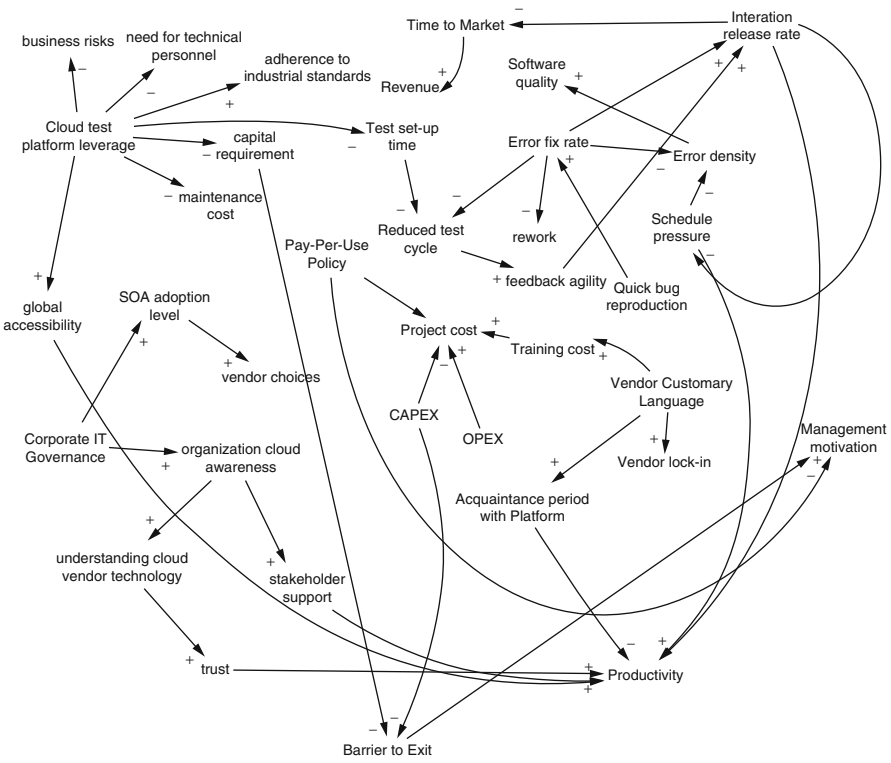

Fig. 8.6 The cloud testing model in software development

vendor technology for its potential adoption. *Stakeholder support* greatly favours *productivity* due to the motivation given by the stakeholders. *Trust* is built in the software development organisation when the cloud vendor technology is fully understood and supported which is extremely vital to maximise productivity in software testing.

The *global accessibility* of the cloud test platform also helps to keep the work going at anytime from anywhere to boost *productivity*. The short test cycles enhance agility and make the entire development cycle reduced, and teams can meet their deadlines better while attaining better productivity. This reduces the schedule pressure and undue tension in teams hence improving productivity.

The only degenerative factor into this variable is *Acquaintance Period of New Platform*. This is the time it takes to get used to the new interface as well as the *vendor customary language* (*DSL*) of the cloud test provider. The time spent to learn the customary language reduces the time that could actually be spent performing testing, hence reducing team productivity.

8.4.2 Project Cost

Due to the reduction on infrastructure and other test resource requirements, costs on the organisation shift from *capital expenditure* to *operational expenditure* needed to keep testing activities running. The *pay-per-use policy* also significantly reduces the costs that would otherwise be spent as a lump sum on the long-term usage purpose. The cost related to the cloud vendor platform language, i.e. *training cost*, however adds to the *project cost*. This includes the opportunity cost as well as the wages of adopting the cloud vendor's platform.

8.4.3 Software Quality

The omnipresent visibility of the project and detected bugs makes it easy to easily fix the bugs without any geographical time constraints. The ability to have snapshots of the detected defects also makes it easy to configure the application to reproduce the bugs for fixation. This immensely improves the *rate of error fixation* and consequently the *error density* in the software. With these reduced effects, higher *software quality* and reliability is achieved. This is further hypothesised to improve the customer satisfaction leading to increased software sales leading to increase in revenue.

8.5 Conclusion

Cloud testing is relatively new in the industry but gradually growing popularity particularly in the application of performance and load testing. In a number of ways, cloud testing enhances the "agility" of the testing process in terms of the reduced development cycle, improved quality and faster ROI. Cloud testing, when integrated well into software projects, accounts for vast savings due to the avoidance of total cost of ownership (TCO) and also helps to reduce TTM. This invention is however not a silver bullet and has a tangible number of risks, particularly security-wise, when companies have to put vital information in the cloud to create a production environment for testing.

There are steps that need to be ensued to determine its profitability, relevance and alignment with an organisation's goals. A unique model has been presented in this chapter which includes all the variables in a cloud test environment – including the risk factors, benefits and causal effects of decisions taken in an organisation adopting cloud testing. Cloud testing is steadily making its mark and software teams are now realising the benefits of testing their software applications in the cloud. Based on the present trend of the pervasiveness of cloud testing, it is expected that cloud testing will have a monumental indelible impact on software testing and development over the next few years.

References

1. Barry, C.: 70+ ways to reduce costs, increase productivity and improve customer service. http://www.fcbco.com/Portals/163466/docs/FCBCO-Cost-Reduction (2008). Accessed 4 May 2012
2. Bykov, S., Geller, A., Kliot, G., Larus, J.R., Pandya, R., Thelin, J.: Orleans: cloud computing for everyone. In: Proceedings of the 2nd ACM Symposium on Cloud Computing, SOCC'11, pp. 16:1–16:14. ACM, New York (2011)
3. Salkever, A.: 5 ways to protect against vendor lock-in in the cloud. http://gigaom.com/2011/09/24/5-ways-to-protect-against-vendor-lock-in-in-the- cloud/ (2011). Accessed 23 Feb 2012
4. Linthicum, D.S.: Cloud Computing and SOA Convergence in Your Enterprise. Addison-Wesley Professional, New York (2009)
5. Jun, W., Meng, F.: Software testing based on cloud computing. In: 2011 International Conference on Internet Computing Information Services (ICICIS), pp. 176–178. IEEE, Los Alamitos (2011)
6. Weidong, F., Yong, X.: Cloud testing: the next generation test technology. In: 10th International Conference on Electronic Measurement Instruments (ICEMI), pp. 291–295. IEEE, Piscataway (2011)
7. David, J.S., Schuff, D., St Louis, R.: Managing your total IT cost of ownership. Commun. ACM **45**(1), 101–106 (2002)
8. Ebbers, M.: Cloud Computing: Save Time, Money and Resources with a Private Test. IBM Redbooks. Raleigh, North Carolina, USA (2009)
9. Kantarcioglu, M., Bensoussan, A., Hoe, S.: Impact of security risks on cloud computing adoption. In: 49th Annual Allerton Conference on Communication, Control, and Computing (Allerton), pp. 670–674. IEEE, Piscataway (2011)
10. Mountjoy, J.: An introduction to Force.com Apex Code. http://wiki.developerforce.com/page/An_Introduction_to_Apex_Code_Test_Methods (2012). Accessed 2 Mar 2012